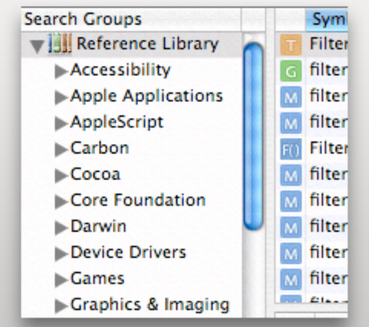
Adopting Xcode

A Hands-On Tour

Session 402

These are confidential sessions—please refrain from filming or taking pictures.



Development Tools

Adopting Xcode

A Hands-On Tour

Chris Espinosa
Manager, DT Productivity
cde@apple.com

http://developer.apple.com/wwdc2005/

Introduction

- Xcode is a full-function development environment
  - Project management
  - Code editor
  - Build system
  - Integrated debugger
- The Good News

  Whatever you're used to, Xcode probably has something like what you want
- The Bad News

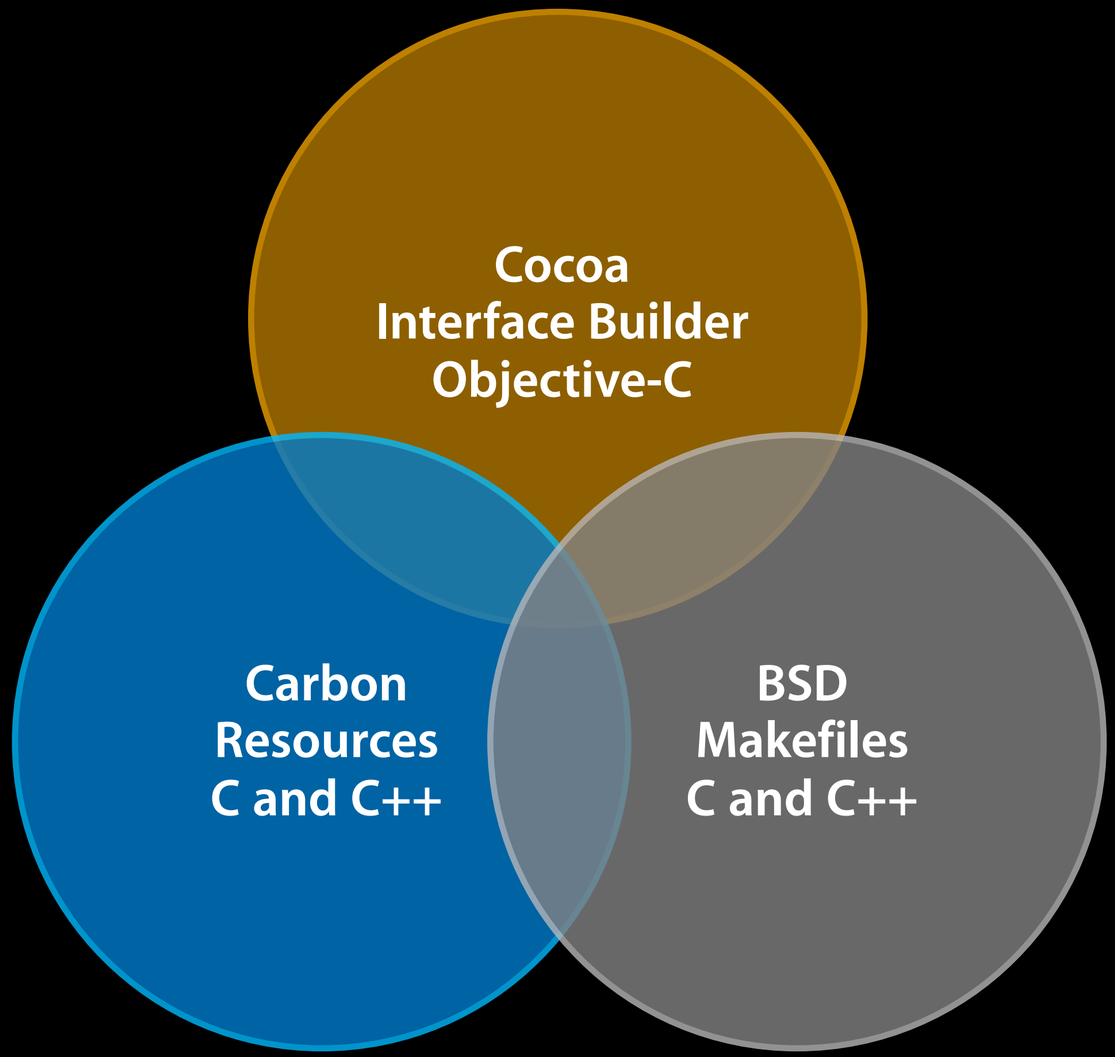
  Whatever you're used to, Xcode's way of doing it may be different

What You'll Learn

- The basics of working with Xcode
- Similarities and differences between Xcode and other IDEs
- "Mental mapping" of CodeWarrior, Visual Studio, and makefile concepts to Xcode

Technology Framework

When Worlds Collide



Steps in Adopting Xcode

- Bring in your existing projects
- Find your way around
- Fix up your project so it builds
- Use Xcode features to simplify your project

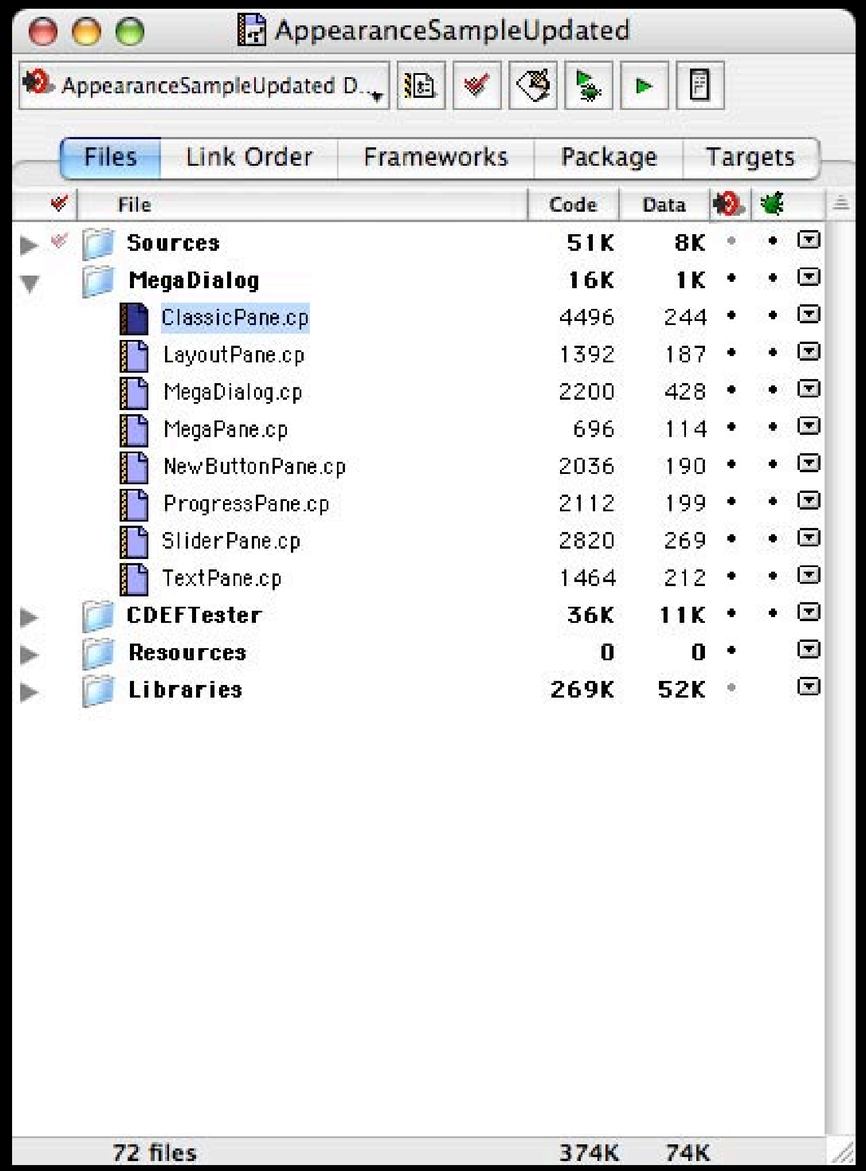
Bringing Your Project Into Xcode

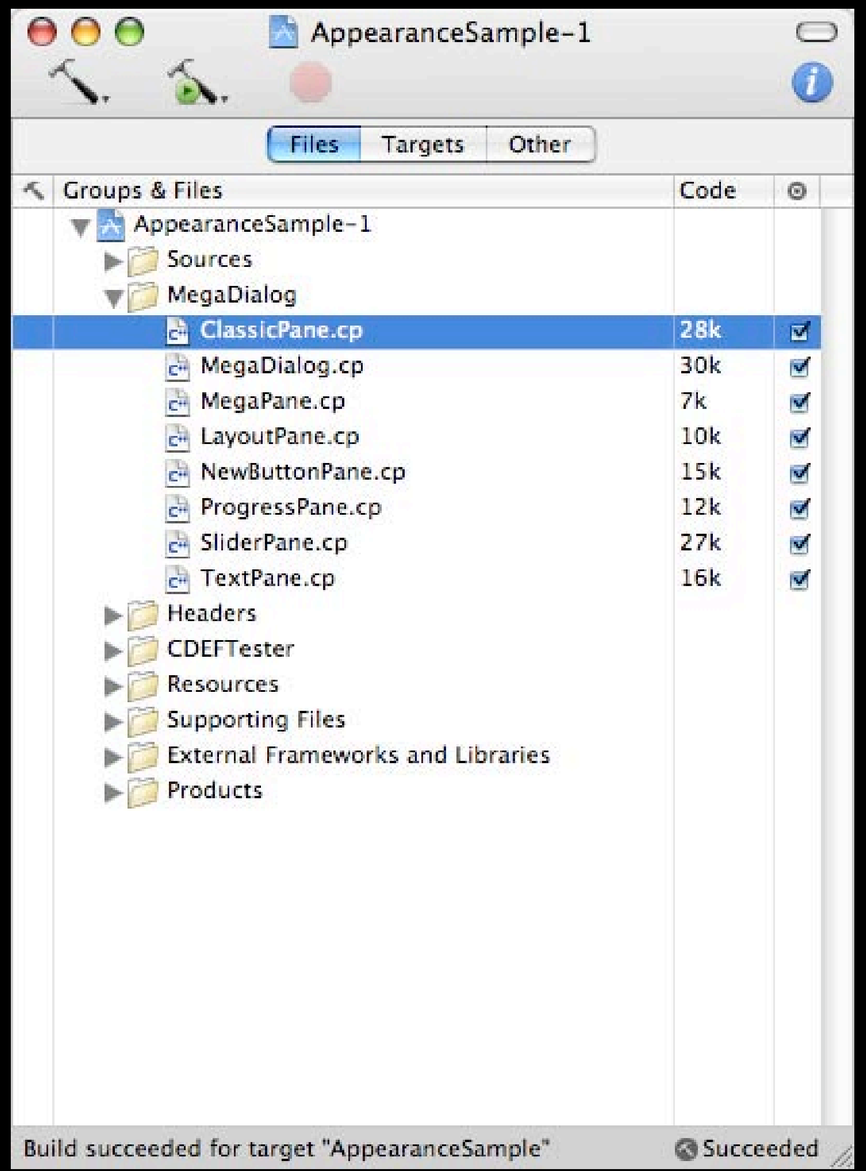
- Use CodeWarrior 9.5 on Tiger to import into Xcode 2.0 or 2.1
  - Note: Tiger headers have notable changes from Panther
- Needless to say, project must be Cocoa or Carbonized
- Make the migration to Mach-O
  - CFM is not supported for Intel or ppc64
- Prepare your code for gcc 4.0
  - See the gcc 4.0 Release Notes for important C++ conformance details

❶ Getting to Know Groups and Files

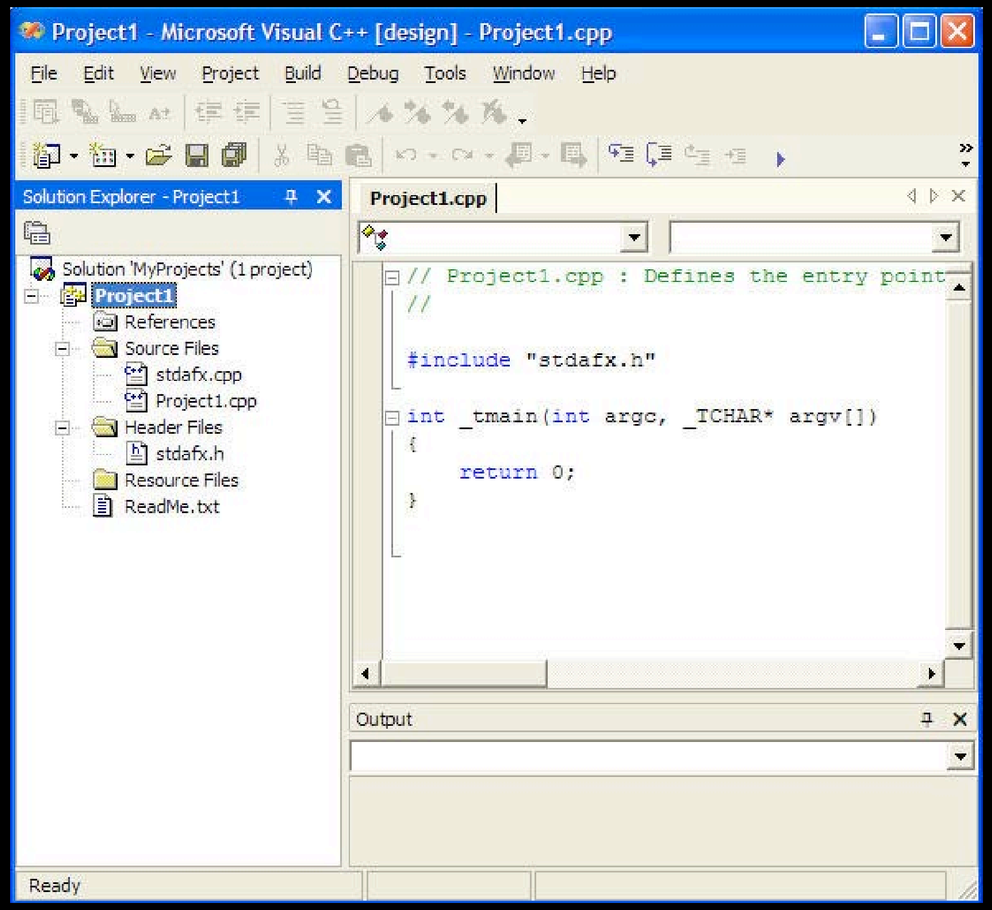
http://developer.apple.com/wwdc2005/

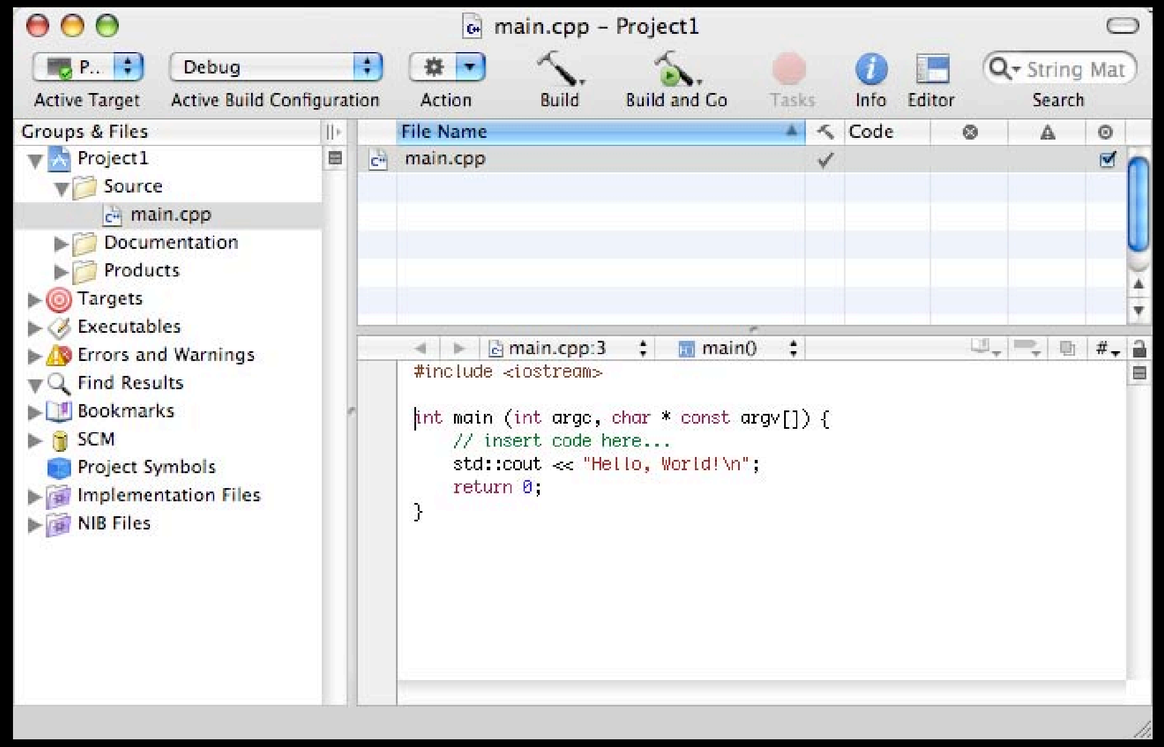
❶ Getting to Know Groups and Files





❶ Getting to Know Groups and Files





❶ Groups and Files Hands-On

http://developer.apple.com/wwdc2005/

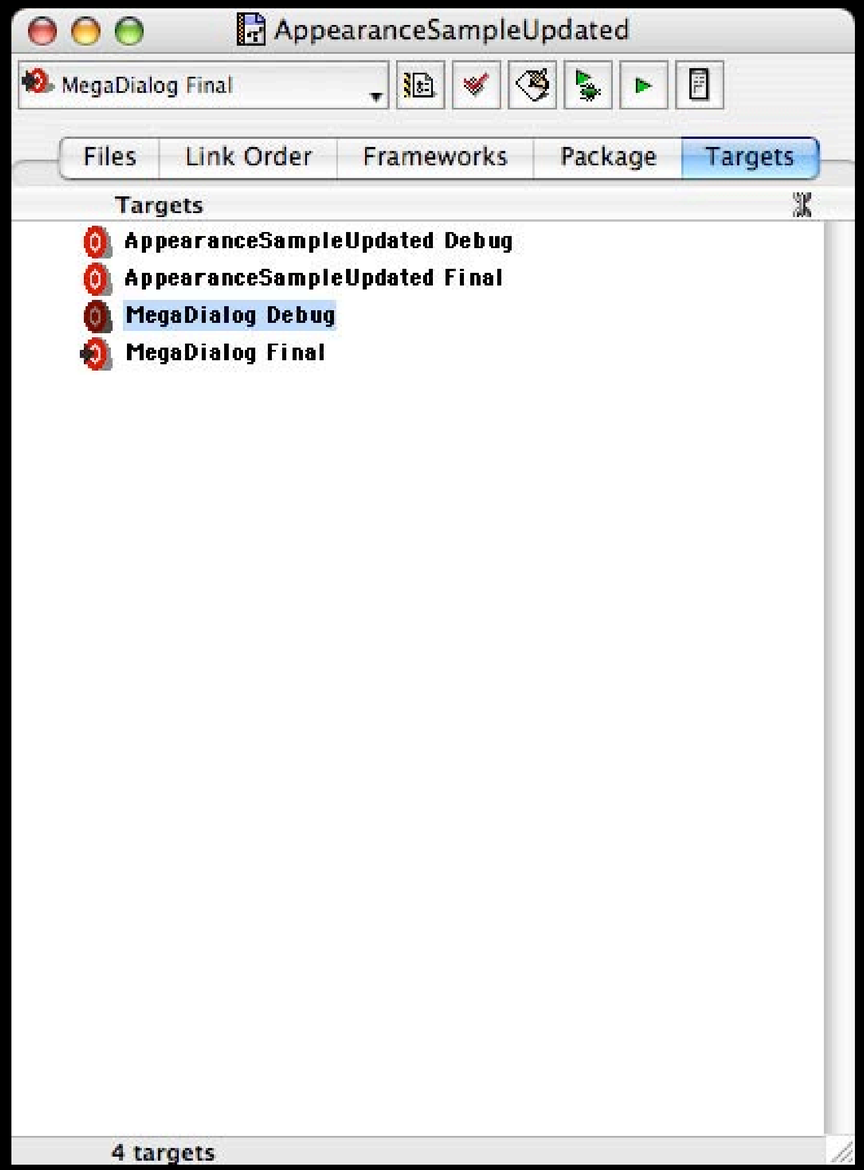
❶ Groups and Files: What's Different

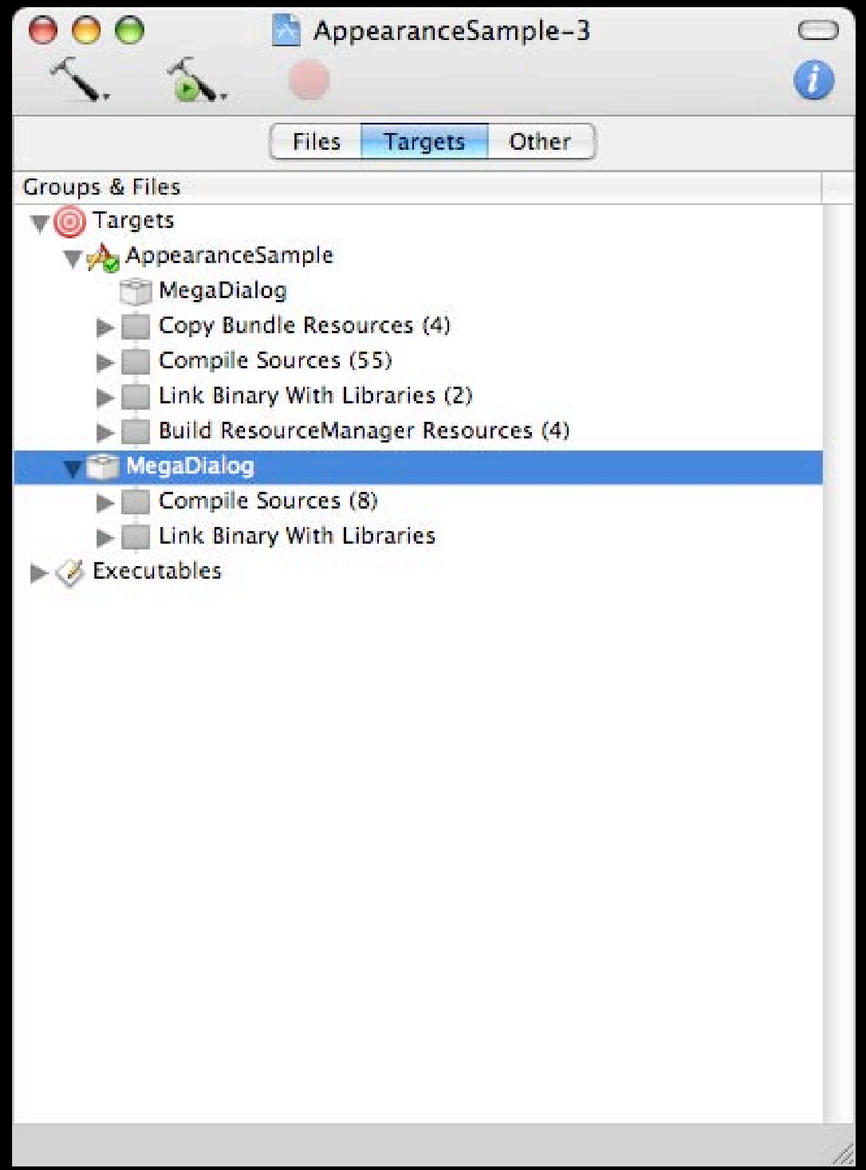
- Xcode offers a choice of UI layouts
- All project files are in the Xcode Groups and Files list
  - Headers
  - Nibs
  - Auxiliary files like .exp, Info.plist sources, prefix files, etc.
- Build and Debug windows are project-specific

❷ Targets and Files

http://developer.apple.com/wwdc2005/

❷ Targets and Files





❷ Targets and Files Hands-On

http://developer.apple.com/wwdc2005/

❷ Targets and Files: What's Different

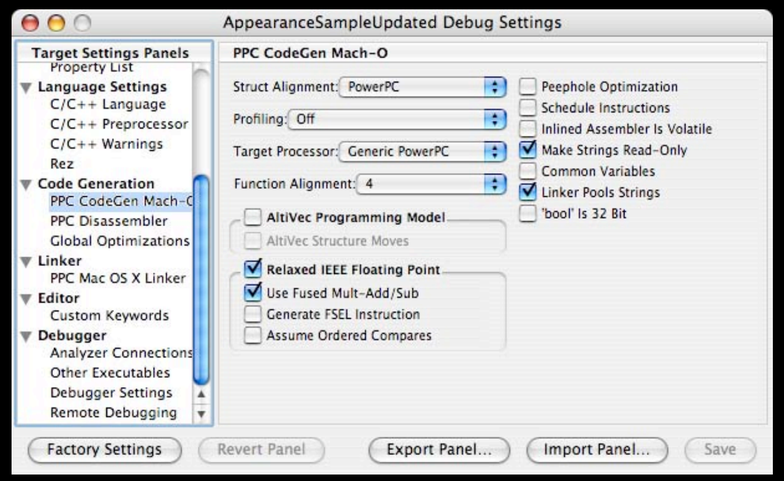
- Set dependencies by drag and drop into Build Phases
- Set link order in Compile Sources build phase

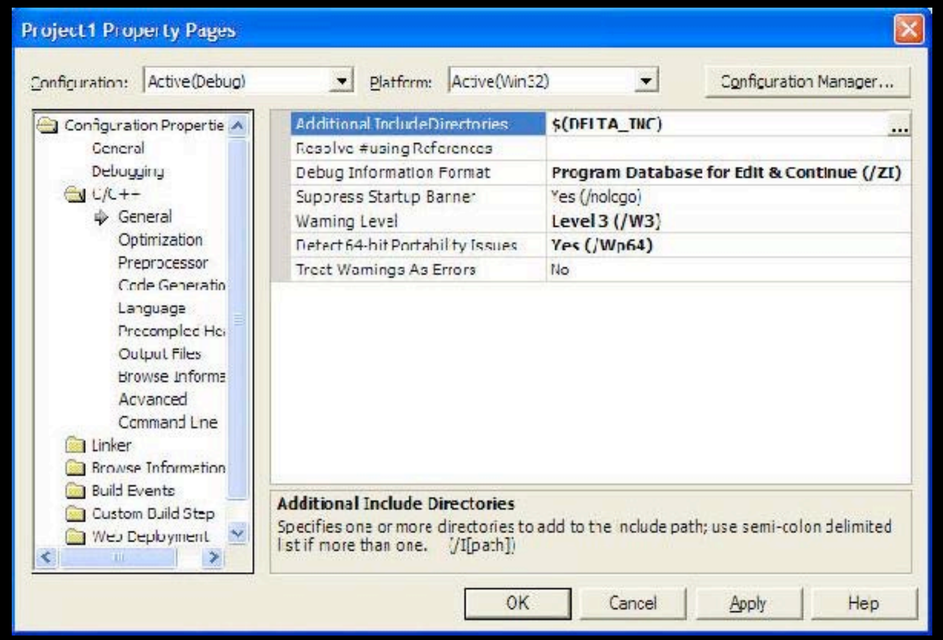
❸ Configurations

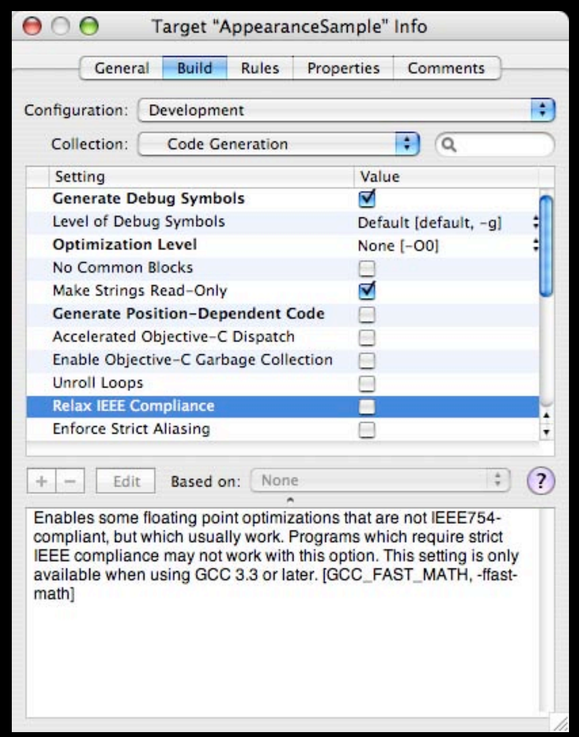
http://developer.apple.com/wwdc2005/

❸ Configurations

- Use configurations instead of creating multiple parallel targets
- Target build settings are set per-configuration
- Powerful searching and filtering helps you find the setting you want







❸ Configurations Hands-On

http://developer.apple.com/wwdc2005/

❸ Configurations: What's Different

Pretty much everything

- You don't need to worry about keeping your Debug and Release targets in synch
- You can easily see what's unique about your Debug build vs. your Release
- Multiple targets can inherit shared settings from the project level

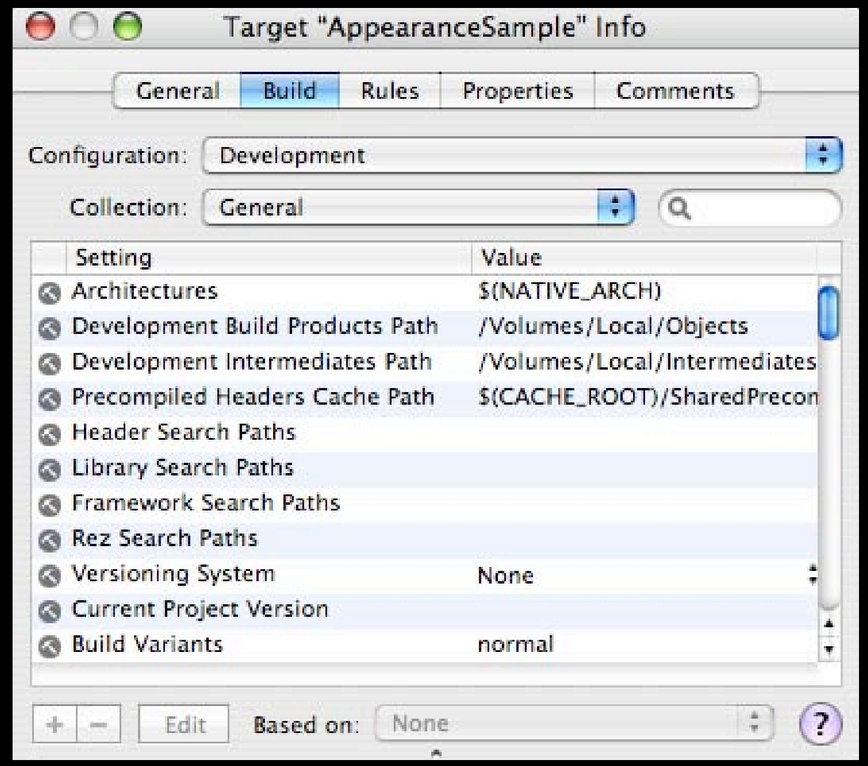
❹ Build Settings

http://developer.apple.com/wwdc2005/

❹ Build Settings

- Not just check boxes and pop-ups
- Values passed on the command line to Unix-based build tools
- Powerful environment-variable substitution

```
TOOL=           attrtest
CC=             cc
CFLAGS=         -fpascal-strings \
                -Wno-four-char-constants \
                -I- \
LIBRARIES=      -framework AppleEvents
```



❹ Build Settings Hands-On

http://developer.apple.com/wwdc2005/

❹ Build Settings: What's Different

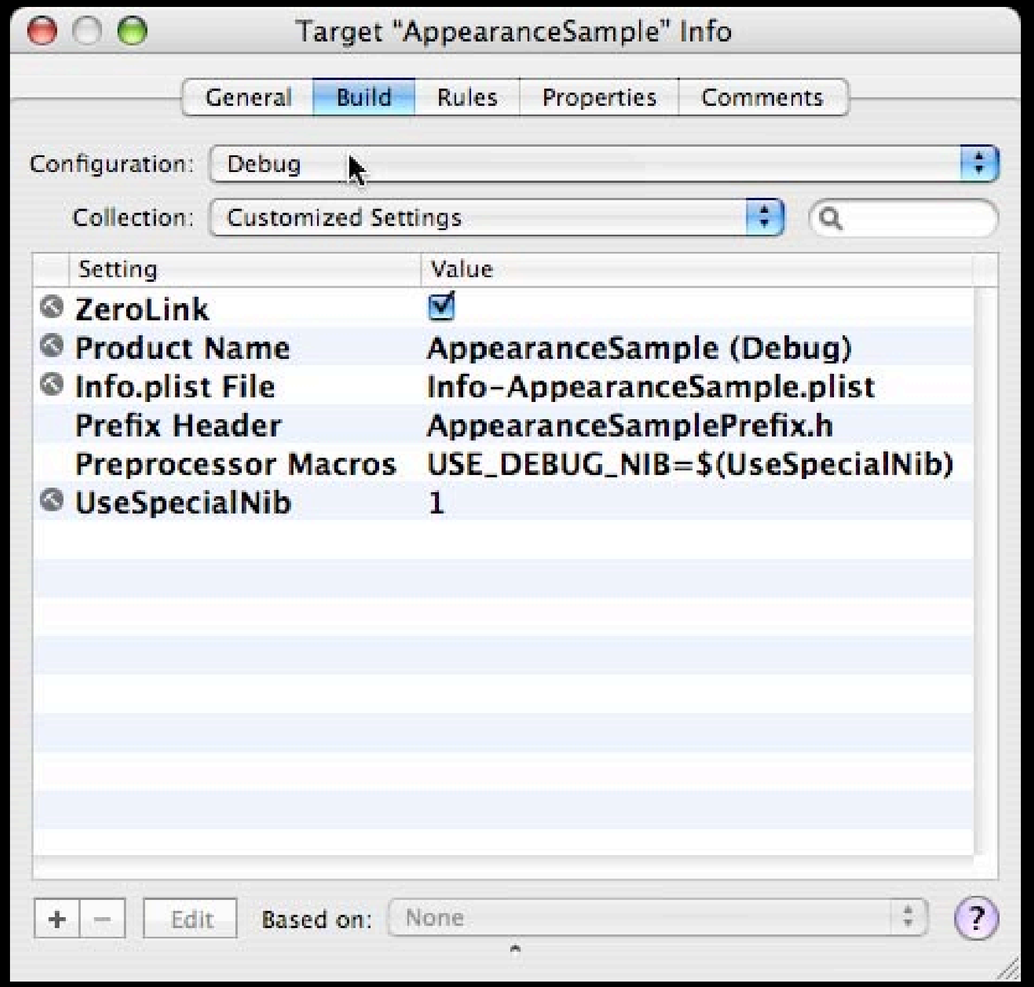
- You can see all settings in one pane
- Use searching and filtering to find your setting, not memorization
- You can base a build setting on the value of another setting with $(SETTING_NAME)
- You can define custom settings to pass to any tool or script
- You can define hierarchies of settings, rather than hardcoding everything per target

5 Customization

http://developer.apple.com/wwdc2005/

❺ Customization

- Your code can be conditionalized per-configuration
- Code, resources, plists, and shell script build phases can use build setting substitution



❺ Customization Hands-On

http://developer.apple.com/wwdc2005/

❺ Customization: What's Different

- No equivalent in CodeWarrior
- More like a GUI to makefiles

❻ Cross-Development

http://developer.apple.com/wwdc2005/

❻ Cross-Development

- Build a Universal Binary of your application

❻ Cross-Development Hands-On

http://developer.apple.com/wwdc2005/

❻ Cross-Development: What's Different

- Universal isn't UniversalInterfaces
- Don't use system-provided headers
- Always use SDKs

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

Power Programming with Xcode	Session 403 Wednesday 5:00PM
Project Management Mastery with Xcode	Session 408 Friday 2:00PM
Moving to gcc 4.0	Session 404 Wednesday 5:00PM

Related Labs and Feedback

Xcode Support	Lab 4000 Tuesday 2:00-5:00PM
Xcode Support	Lab 4001 Wednesday 9:00-12:00PM
Xcode Support	Lab 4002 Thursday 9:00-12:00PM
Developer Tools Feedback Forum	FF410 Friday 5:00PM

Who to Contact

Matthew Formica
Developer Tools and Cocoa Evangelist
mformica@apple.com

xcode-users Discussion List
Community-Based Support for Xcode Users
xcode-users@lists.apple.com

Q&A

Matthew Formica
Developer Tools and Cocoa Evangelist
mformica@apple.com

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference